

### Лабораторная работа 1.3 Xenomai Real Time Linux — Использование семафоров.

**Цель работы:** Знакомство с Xenomai real time Linux, запуск простых задач и использование семафоров.

**Аппаратное и программное обеспечение:** PC, ОС Linux, Virtual Box, Xenomai virtual box image.

#### Порядок выполнения работы:

1. Работа выполняется на локальной или удаленной виртуальной машине с установленным Xenomai. В случае выполнения работы на локальной виртуальной машине запустите VirtualBox и активируйте виртуальную машину с установленным Xenomai. Для входа в систему используйте имя пользователя **root** и пустой пароль. В случае использования удаленной виртуальной машины запустите терминал и введите команду:

**ssh -l root ip\_адрес\_удаленной\_машины**

2. Если необходимо, создайте рабочий каталог, в качестве имени каталога необходимо указать фамилии выполняющих работу, например:

**mkdir Ivanov\_Petrova**

3. В рабочем каталоге необходимо создать файл lab13.c содержащий код программы запуска двух задач. Пример кода:

```
#include <stdio.h>
#include <signal.h>
#include <unistd.h>
#include <sys/mman.h>

#include <native/task.h>
#include <native/timer.h>
#include <native/sem.h>

#include <rtdk.h>

#define ITER 10

static RT_TASK t1;
static RT_TASK t2;

int global = 0;

void taskOne(void *arg)
{
    int i;
    for (i=0; i < ITER; i++)
    {
        rt_printf("I am taskOne and global = %d.....\n", ++global);
    }
}

void taskTwo(void *arg)
{
    int i;
    for (i=0; i < ITER; i++)
    {
```

```

    rt_printf("I am taskTwo and global = %d-----\n", --global);
}
}

int main(int argc, char* argv[])
{
    /* Perform auto-init of rt_print buffers if the task doesn't do so */

    rt_print_auto_init(1);

    /* Avoids memory swapping for this program */

    mlockall(MCL_CURRENT|MCL_FUTURE);

    /* create the two tasks */

    rt_task_create(&t1, "task1", 0, 1, 0);
    rt_task_create(&t2, "task2", 0, 1, 0);

    /* start the two tasks */

    rt_task_start(&t1, &taskOne, 0);
    rt_task_start(&t2, &taskTwo, 0);

    return 0;
}

```

Выполните компилирование программы. Запустите программу (подробные инструкции по компилированию и запуску программ см в методических указаниях к лабораторной работе 1.1).

4. Добавьте в код программы семафор. Используя семафор измените ход выполнения программы — заблокируйте доступ задаче **TaskOne** к глобальной переменной **global** до тех пор, пока не выполнится задача **TaskTwo**. Выполните компилирование и запустите программу.

### Приложение 1.3.1

Использование семафоров в Xenomai

```
#define SEM_INIT 1 // Установка начального значения семафора
```

```
#define SEM_MODE S_FIFO // Установка режима работы семафора
```

```
RT_SEM sem_desc; // Объявление семафора
```

```
rt_sem_create(&sem_desc,"MySemaphore",SEM_INIT,SEM_MODE); // Создать семафор...
```

```
rt_sem_p(&sem_desc,TM_INFINITE); // Уменьшение счетчика семафора, если счетчик = 0 то включается блокировка процесса
```

```
rt_sem_v(&sem_desc); // Увеличение счетчика семафора
```

```
rt_sem_broadcast (&sem_desc); // Общий сигнал (разблокирование всех задач)
```

```
rt_sem_delete(&sem_desc); // Удаление семафора
```

**Источники информации:**

[www.xenomai.org](http://www.xenomai.org)

[www.google.ru](http://www.google.ru)